

Computing Metric Distances Between Shapes and the Euler-Poincaré Equations of Computational Anatomy

J. Tilak Ratnanather

Whitaker Biomedical Engineering Institute

<http://cis.jhu.edu>



“Glorious Fluidity of Fluids”

Sir James Lighthill

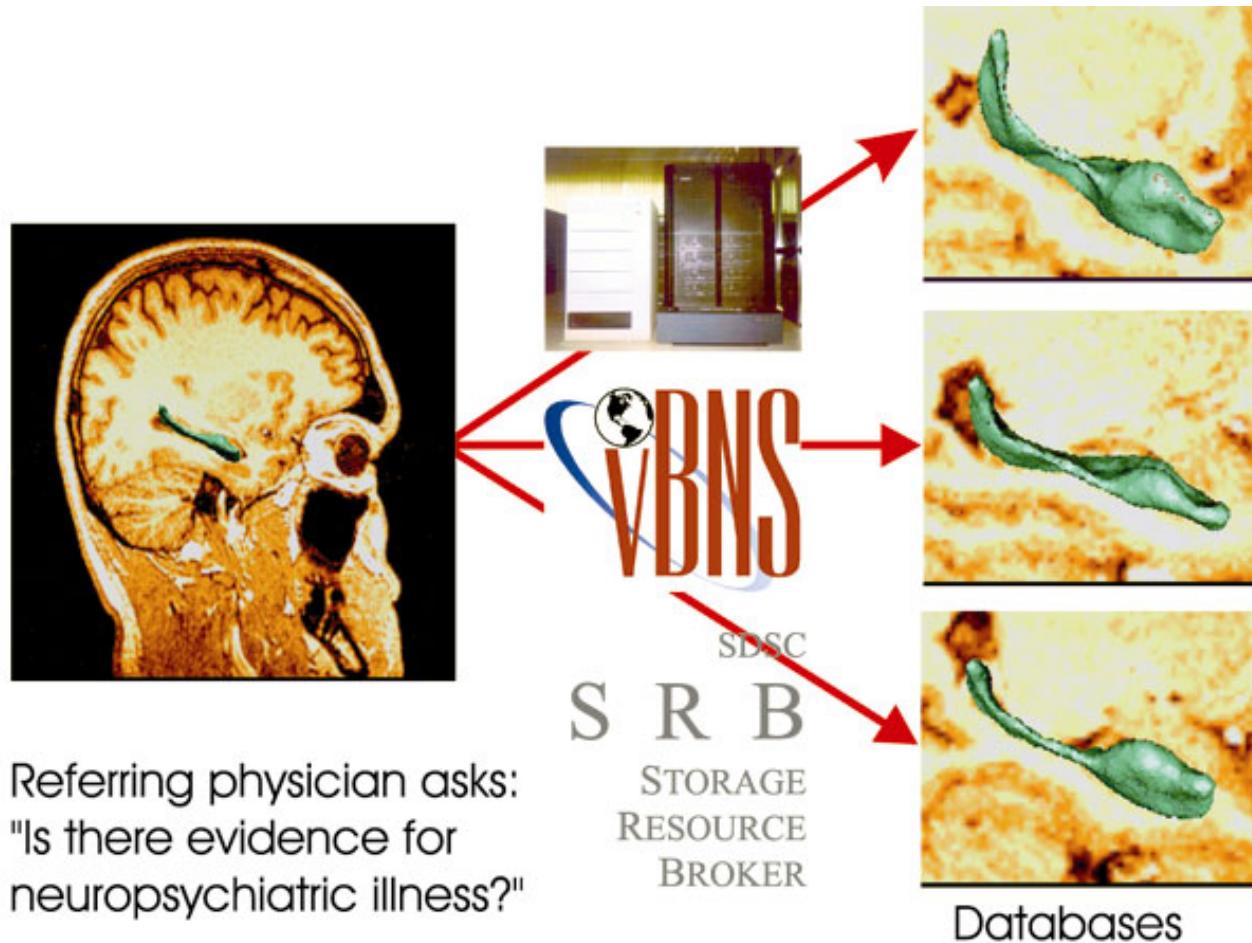
Outline

- Metric Distance Summary
- Computing Metric Distances: Beg Gradient Method
- Greedy Algorithm: 1D MATLAB simulation
- Numerical Simulations on IBM RS6000/SP supercomputer
- Euler-Poincaré Equations of Computational Anatomy

- Faisal Beg, Ph.D. Candidate (Beg Gradient Method)
- Michael Miller, Laurent Younes & Alain Trouv  (theory)
- Kate Johnson (BME/Math Sciences double major)
- Steve Baigent (UCL – Euler Poincar  equations of CA)

Supported by NSF NPACI, NIH NCRR, MAA

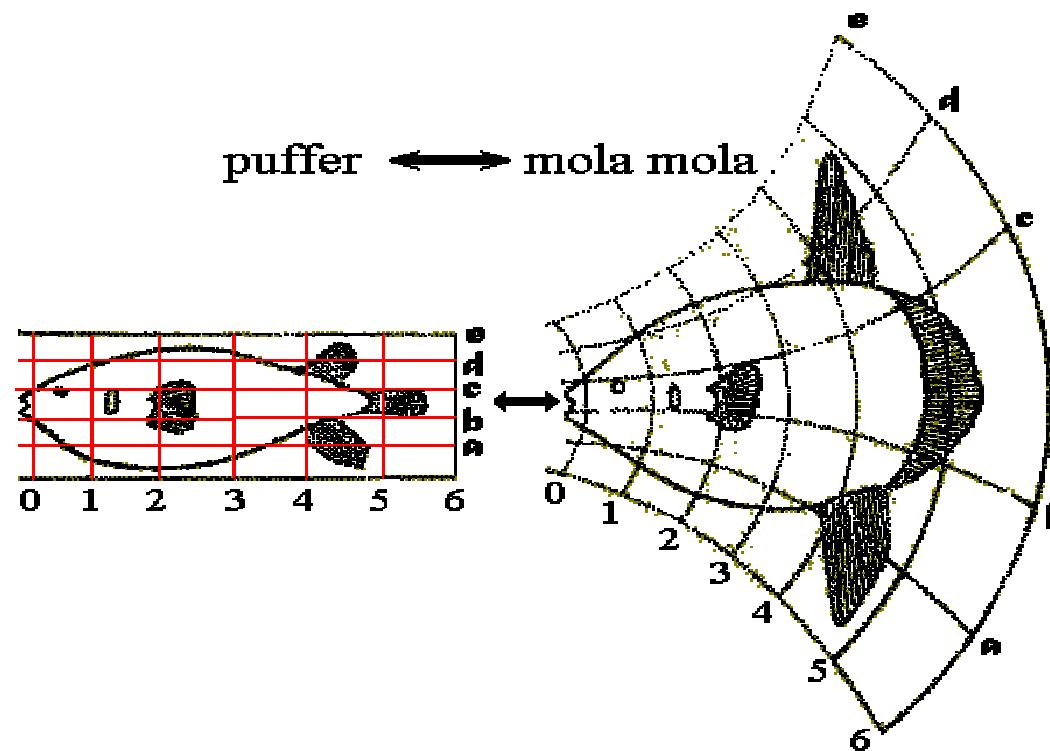
NSF/NPACI Neuroscience Thrust Goal



Compute metric distances on federated anatomical
image databases across high speed network

"In a very large part of morphology, our essential task lies in the comparison of related forms rather in the precise definition of each; This process of comparison, of recognizing in one form a definite permutation or deformation of another, ... is the Method of Coordinates, on which is based the Theory of Transformations".

D'Arcy Wentworth Thompson, *On Growth and Form*, 1917 (Dover Pub.)



Metric Distance Summary (1)

- Miller, Trouvé, Younes (2002) “On the Metrics and Euler-Lagrange Equations of Computational Anatomy” *Annual Review of Biomedical Engineering*, 4:375-405 plus supplement
(<http://bioeng.annualreviews.org/cgi/reprint/4/1/375>)
- Miller, Younes (2002) “Group Actions, homeomorphisms, and matching: a general framework” *Int. J Comput. Vision* 41:61-84
(<http://www.kluweronline.com/issn/0920-5691>)
- Younes (2000) “Deformations, Warping and Object Comparision: A tutorial” (<http://www.cmla.ens-cachan.fr/Utilisateurs/younes/eccvTutorial/tutorialWarping.ps.gz>)

Metric Distance Summary (2)

- **anatomy is a Grenander deformable template: an orbit generated from a template under groups of diffeomorphisms**
- **construct metric spaces from geodesics connecting one anatomical structure to another in the orbit**

Images are functions

$I(x), x \in X$

Anatomy is all images generated
from a template

$$ANATOMY = \underbrace{G \cdot I_{temp}}_{group-action} = \left\{ \underbrace{I_{temp} \circ \phi}_{function-composition}, \phi \in G \right\}$$

Metric Distance Summary (3)

- Mumford & Vishik (1998) derived E-L equations via calculus of variations
- Miller, Younes & Trouv  (2002) formally derived E-L equations w.r.t. perturbations in group elements
- Beg (2002) developed gradient method to solve E-L equations w.r.t. velocity perturbations
- Ratnanather & Baigent (1999) recast the E-L equations in Euler-Poincar  framework

Web Course (Spring/Summer 2003)

Education link at CIS website for interactive course
on “**Introduction to Metric Pattern Theory**” based on

- Grenander “Elements of Pattern Theory” (JHU Press, 1996)
- Singer “Symmetry in Mechanics: A gentle. Modern Introduction” (Birkhauser, 2001)

Chapters:

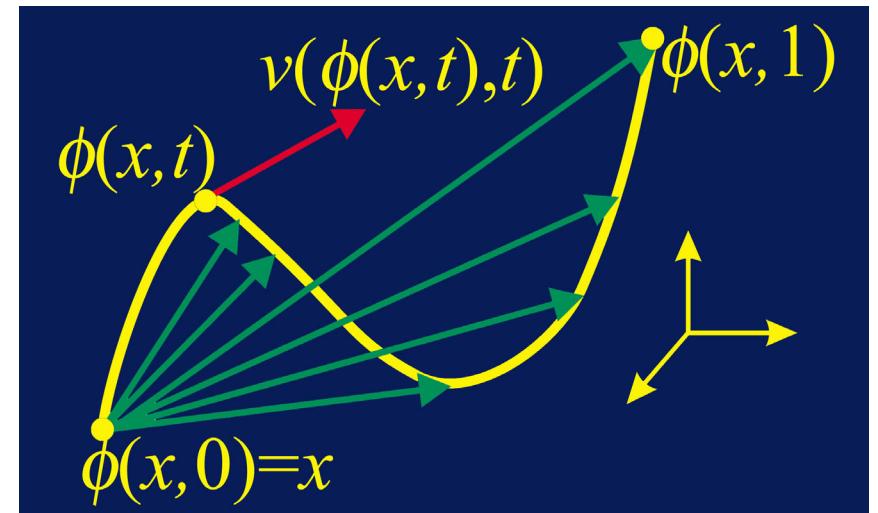
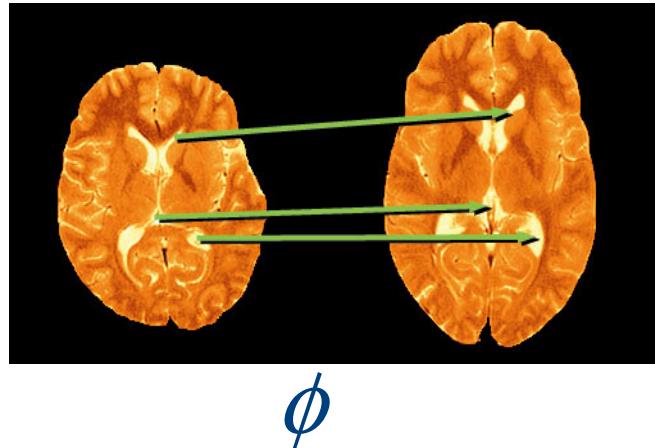
1. **Group Theory**
2. **Matrix Groups**
3. **Lie Groups**
4. **Group Actions, Orbits and Geodesics**
5. **Deformable Templates**
6. **Computing Metric Distances**



**Mathwright via
Project Welcome (MAA)**

Computing the geodesic: problem statement

I_0 : Template I_1 : Target



Problem Statement: Given I_0 and I_1 , compute v such that

$$\arg \min_v \int_0^1 \int_{\Omega} \langle Lv(y, t), Lv(y, t) \rangle dy dt + \int_{\Omega} (I_0(\phi^{-1}(y, 1)) - I_1)^2 dy$$

$$\text{where } \frac{\partial \phi^{-1}}{\partial t}(y, t) = -\nabla_y^t \phi^{-1}(y, t) v(y, t)$$

Beg Ph.D
thesis (2002)

Computing the geodesic: variational approach

$E(v)$ is the energy associated with $v(y,t)$:

$$E(v) = \int_0^1 \int_{\Omega} \langle Lv(y,t), Lv(y,t) \rangle dy dt + \int_{\Omega} (I_0(\phi^{-1}(y,1)) - I_1)^2 dy$$

First variation is:

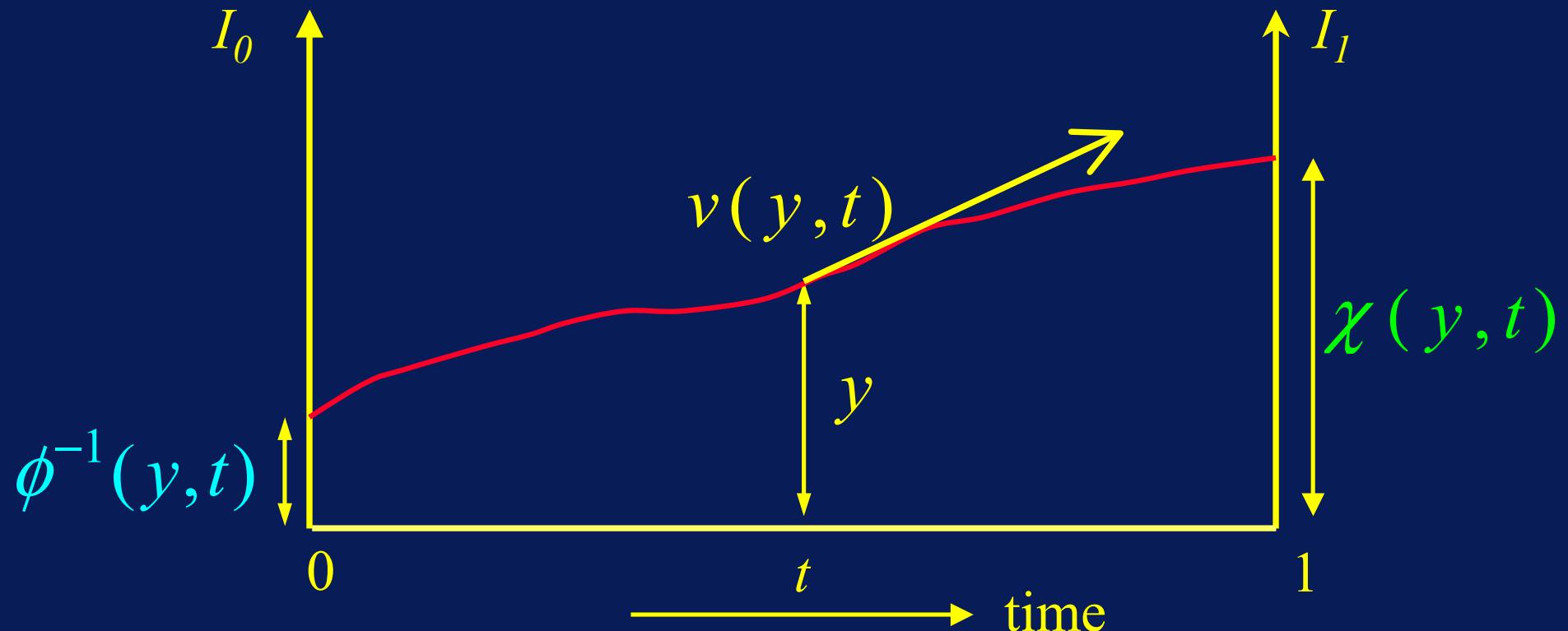
$$E(v + \varepsilon \xi) - E(v) = \varepsilon \int_0^1 \int_{\Omega} \langle \nabla_v E, \xi \rangle dy dt + o(\varepsilon)$$

Solve via Gradient Descent: $v^{n+1} = v^n - \varepsilon \nabla_v E$

Beg Ph.D
thesis (2002)

Computing the gradient descent: schematic

$$\nabla_v E(v) = v(y, t) - (\dot{L}^* L)^{-1} \left(|D\chi(y, t)| D \left(I_0 \circ \phi^{-1}(y, t) \right)^* \left(I_0 \circ \phi^{-1}(y, t) - I_1 \circ \chi(y, t) \right) \right)$$



$$\phi^{-1}(y, 0) = x$$

$$\chi(y, t) = \phi(\phi^{-1}(y, t), 1)$$

Computing the gradient descent: ϕ^1 and χ

- Calculating maps ϕ^1 and χ requires integrating an advection (convection) equation where $g=\phi^1$ or χ is convected by the velocity and is conserved :

$$\frac{dg}{dt} = \frac{\partial g}{\partial t}(y, t) + \nabla_y^t g(y, t) v(y, t) = 0$$

- Use semi-Lagrangian method used in computational fluid dynamics e.g. numerical weather prediction (Côté & Staniforth, 1990)

$$\phi^{-1}(y, t) = \phi^{-1}(y - \alpha, t - \Delta t) \text{ with } \phi^{-1}(y, 0) = y$$

$$\chi(y, t) = \chi(y + \alpha, t + \Delta t) \text{ with } \chi(y, 1) = y$$

$$\frac{\alpha}{2} = \frac{\Delta t}{2} v\left(y - \frac{\alpha}{2}, t - \frac{\Delta t}{2}\right)$$

Computing the gradient descent: $(L^\dagger L)^{-1}$

$$\nabla_v E(v) = v(y, t) - (\dot{L}^\dagger L)^{-1} \left(|D\chi(y, t)| D(I_0 \circ \phi^1(y, t))^* (I_0 \circ \phi^1(y, t) - I_1 \circ \chi(y, t)) \right)$$

- L chosen as $-\alpha \nabla^2 + \gamma I$ with periodic boundary conditions on a discrete lattice of dimension N^3 and so L is self-adjoint i.e. $L = L^\dagger$
- finite difference discretization of $LL^\dagger f = h$ where f and h defined on lattice
- apply Fourier transform to discretized equations to obtain

$$F(k) = \begin{pmatrix} A^{-2} & 0 & 0 \\ 0 & A^{-2} & 0 \\ 0 & 0 & A^{-2} \end{pmatrix} H(k) \text{ where } A = \left(\sum_{i=1}^3 -2\alpha \cos \frac{2\pi k_i}{N} + 6\alpha + \gamma \right)$$

- inverse Fourier transform of F permits us to compute $(LL^\dagger)^{-1} h$ and thus the gradient descent

Computational Algorithm

1. time steps $t_j \in [0,1]$ for $j = 0, \dots, N-1$
2. $k = 1, \dots, K$ initialize $v_k(y, t_j) = 0$, $\phi_k^{-1}(y, t_j) = y$, $\phi_k(\phi_k^{-1}(y, t_j), 1) = y$ and $\nabla_v E(v_k(y, t_j)) = 0 \quad \forall t_j$
3. Calculate $v_{k+1} = v_k - \nabla_v E(v_k)$
4. Reparameterize velocity as constant speed along geodesic
5. $j = N-1, \dots, 0$ compute $\chi_{k+1}(y, t_j)$ and $I_1(\chi_{k+1}(\cdot, t_j))$
6. $j = 0, \dots, N-1$ compute $\phi_k^{-1}(y, t_j)$, $I_0(\phi_k^{-1}(\cdot, t_j))$,
7. $j = 0, \dots, N-1$ compute $D(I_0(\phi_k^{-1}(\cdot, t_j)))$ and $|D(\chi(\cdot, t_j))|$
8. $j = 0, \dots, N-1$ compute $\nabla_v E(v_{k+1})$
9. Compute $\|\nabla_v E(v_{k+1})\|$ until below tolerance
10. Update $E(v_{k+1})$ and re-iterate (i.e go to 3) until $k > K$

Beg Ph.D
thesis (2002)

Christensen “Greedy” Algorithm: Beg implementation

- original large deformation mapping used a “greedy” dynamic programming shooting type algorithm (Christensen et al., IEEE Trans. Image Processing, 5:1435-1447, 1996)
- upper bound on metric distance but computed path is not the shortest path
- however it is fast

1. Initialize: $t_j = 0$, $v(x, t_j) = 0$, $\phi(x, t_j) = \text{Id}$
2. Define $J_t^0(x) = I_0(\phi^{-1}(x, t))$ and $J_t^1(x) = I_1$
3. Solve $Kv_t + b_t = 0$ where $b_t = (J_t^0(x) - J_t^1(x))\nabla J_t^0(x)$
4. Timestep $t_j = t_j + \Delta t$ where $\Delta t = \Delta x / (C \max(|v_{t_j}|))$
5. Update $J_{t_j}^0(x) = I_0(\phi^{-1}(x_j - v_j \Delta t, t))$
6. Repeat 3-5 until $\text{abs}((J_t^0(x) - J_t^1(x)) / J_t^1(x)) < 0.02$

Beg Ph.D
thesis (2002)

1D MATLAB SIMULATION

- 2 Gaussian Shapes (with overlap)
- Greedy Algorithm using MATLAB (short code available at CIS website)
- adjust parameters α, γ
- a real-life application would be “time warping” of two speech signals
- MATLAB implementation of Beg Gradient Method in development



Computing geodesics on JHU-CIS Zeus and UCSD/SDSC Blue Horizon

- ZEUS is a 80 processor IBM RS6000/SP supercomputer. It is a clustered 16-way SMP architecture.



ZEUS at JHU-CIS

- 8 Gbytes of memory/node. Nodes connected via high-speed IBM switch (350 Mb/s, 17 μ s).

- 375 Mhz, 64 bit, pipelined Power3 processors (peak 1.5 Gflops).



BLUE HORIZON at UCSD

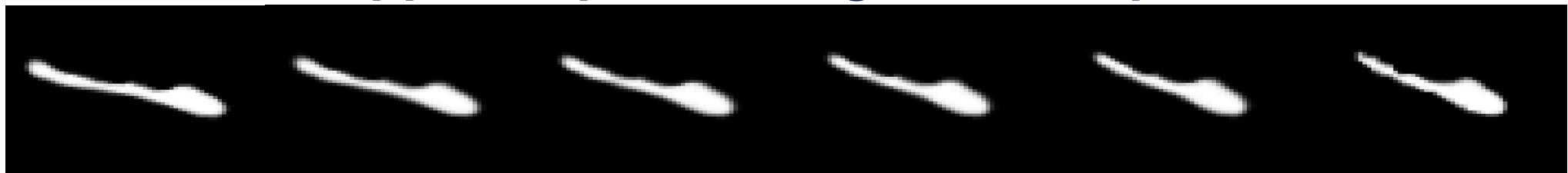
Results of
numerical computation of
metric distances between
shapes

by

M. Faisal Beg

Ph.D. thesis (2003)

2D Hippocampus: Young to Schizophrenia



Young

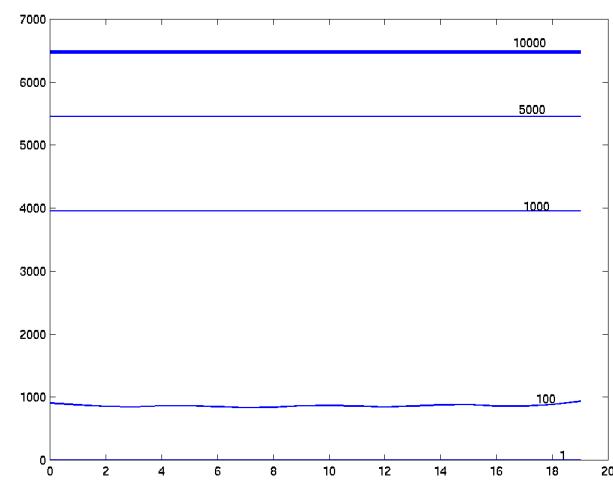
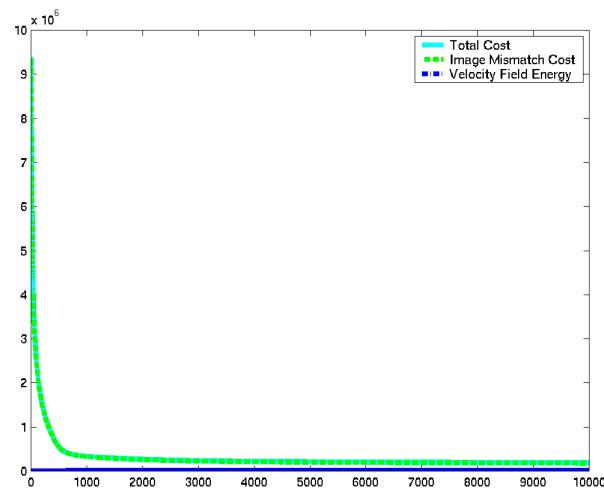
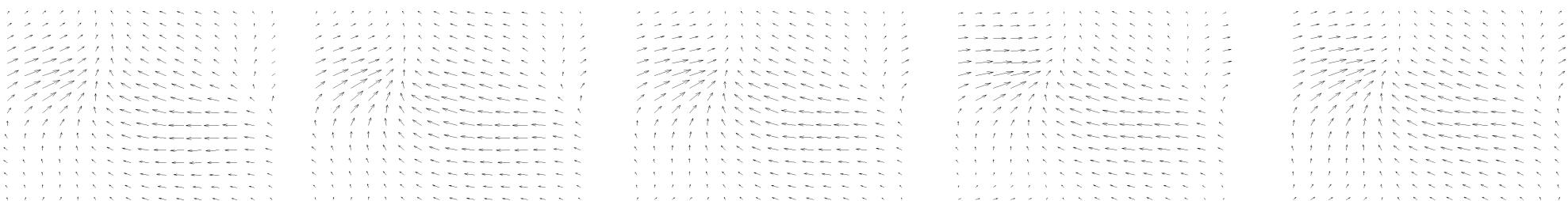
48.27

88.50

128.72

160.91

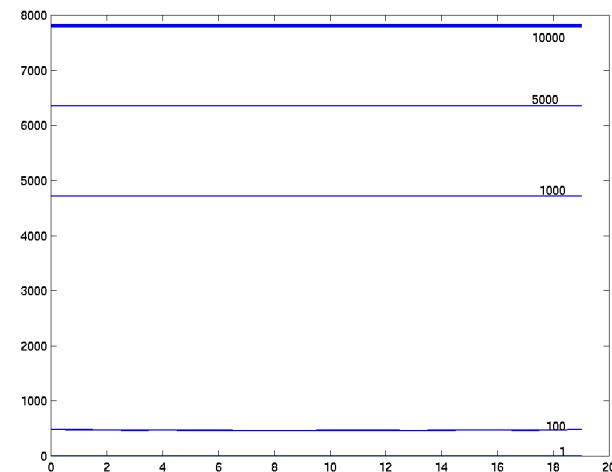
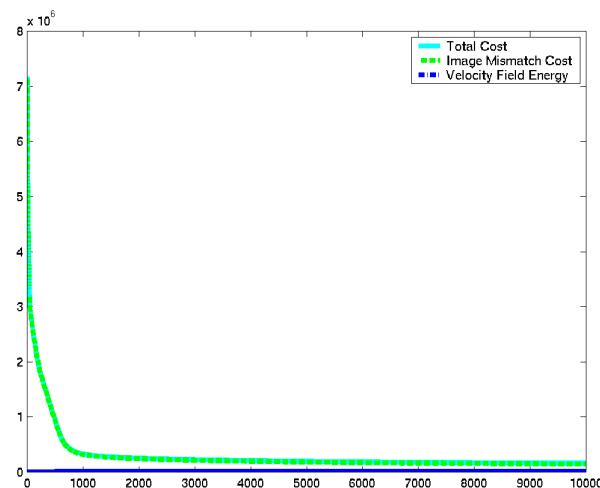
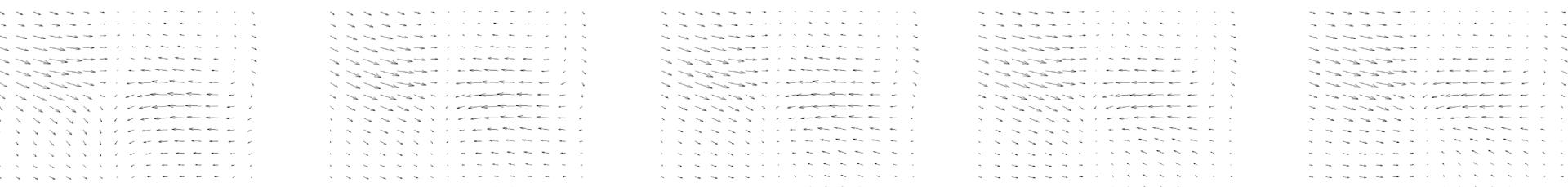
Schizophrenia



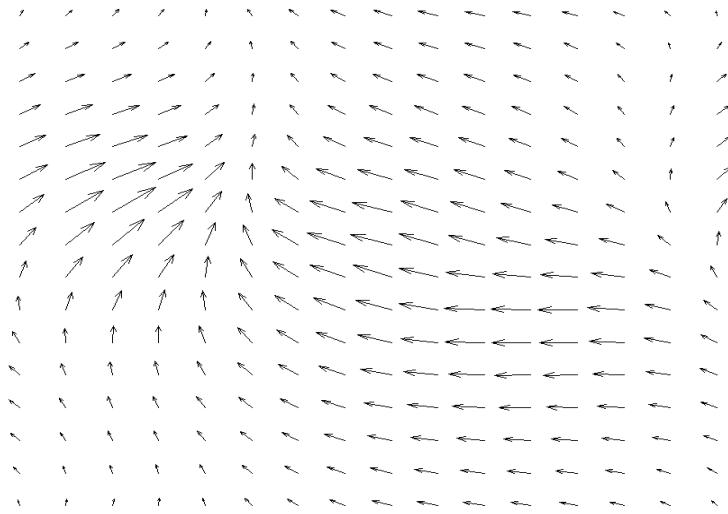
2D Hippocampus: Young to Alzheimer's



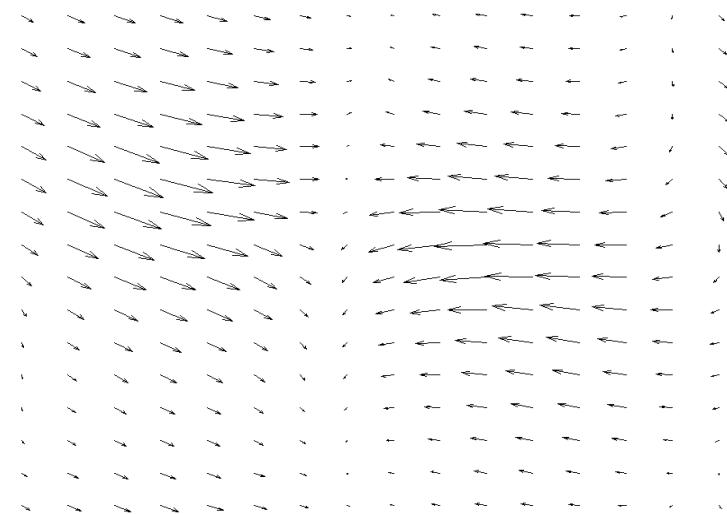
Young 53.00 97.17 141.35 176.68 Alzheimer's



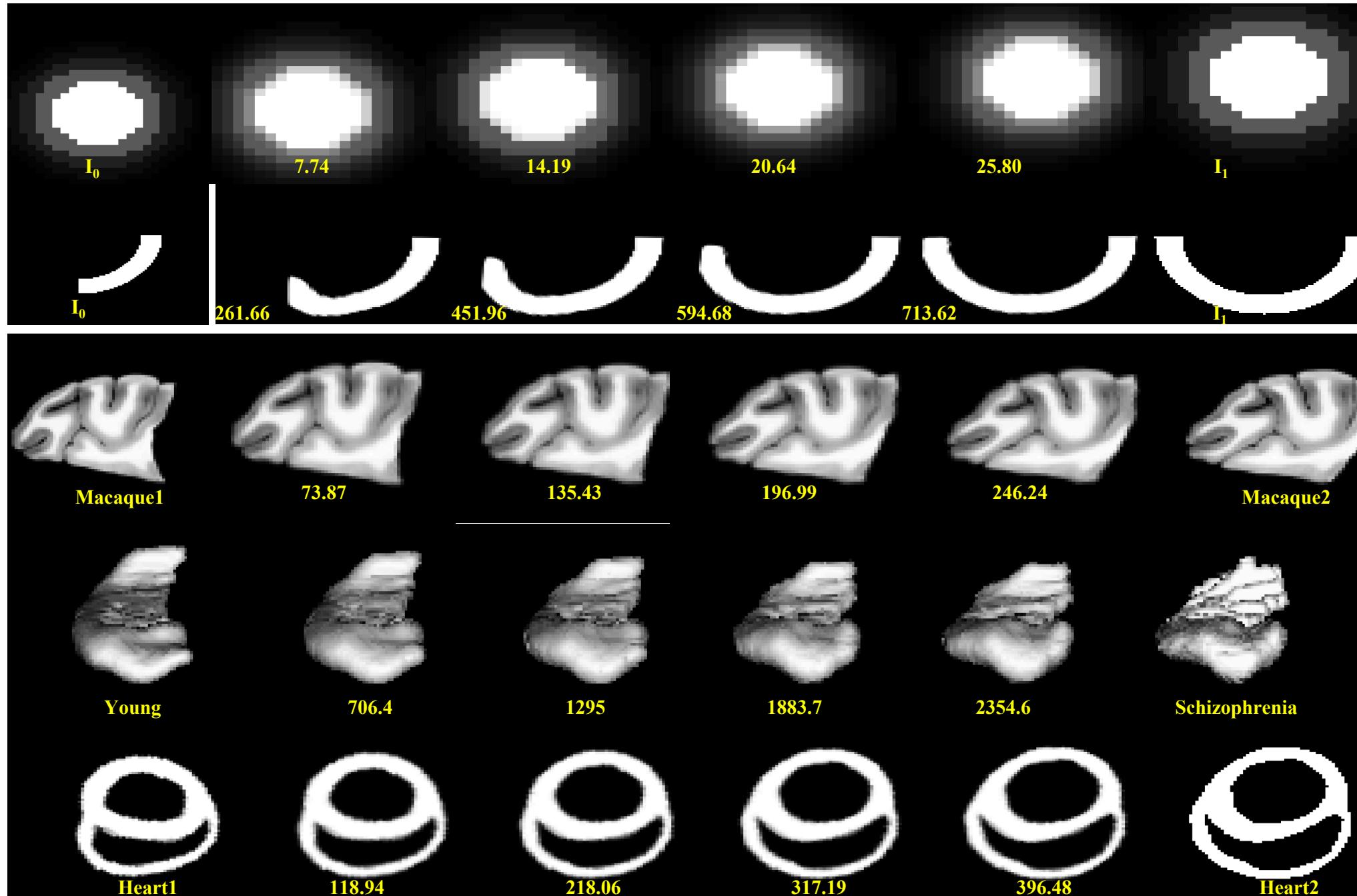
Velocity vectors different on geodesics in Schizophrenia and Alzheimer's



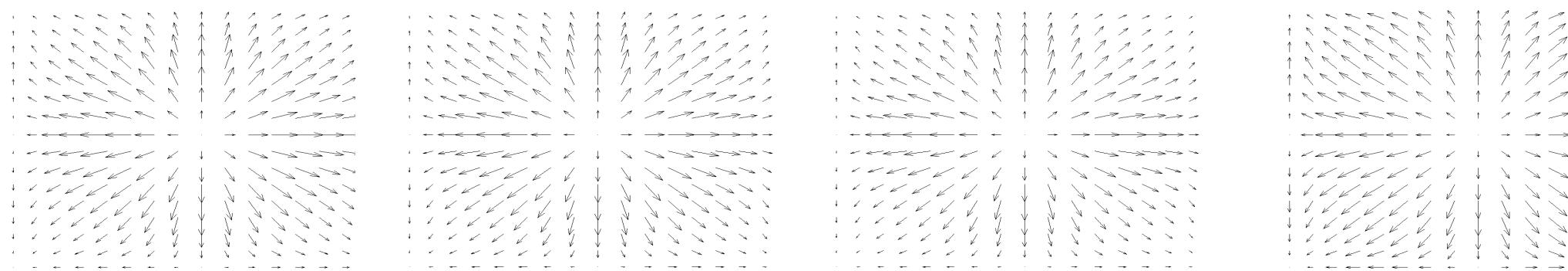
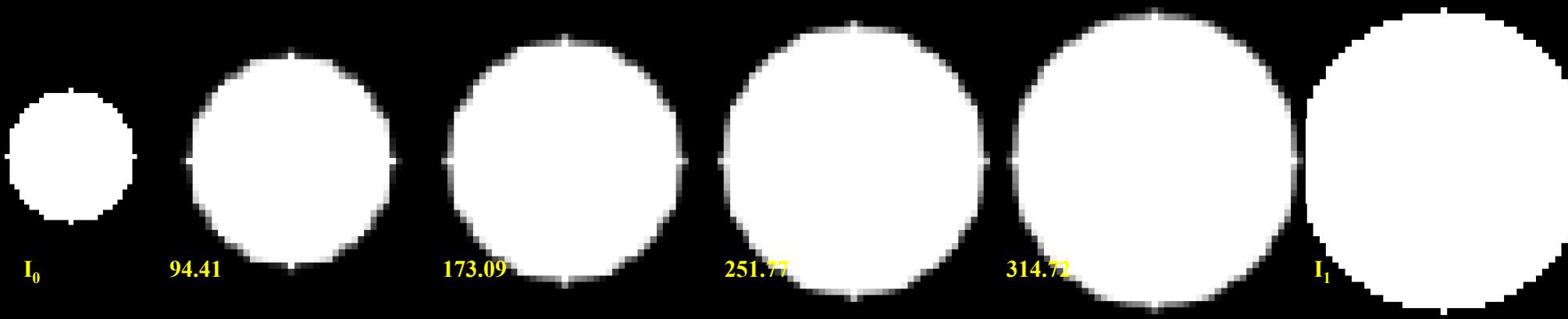
Schizophrenia



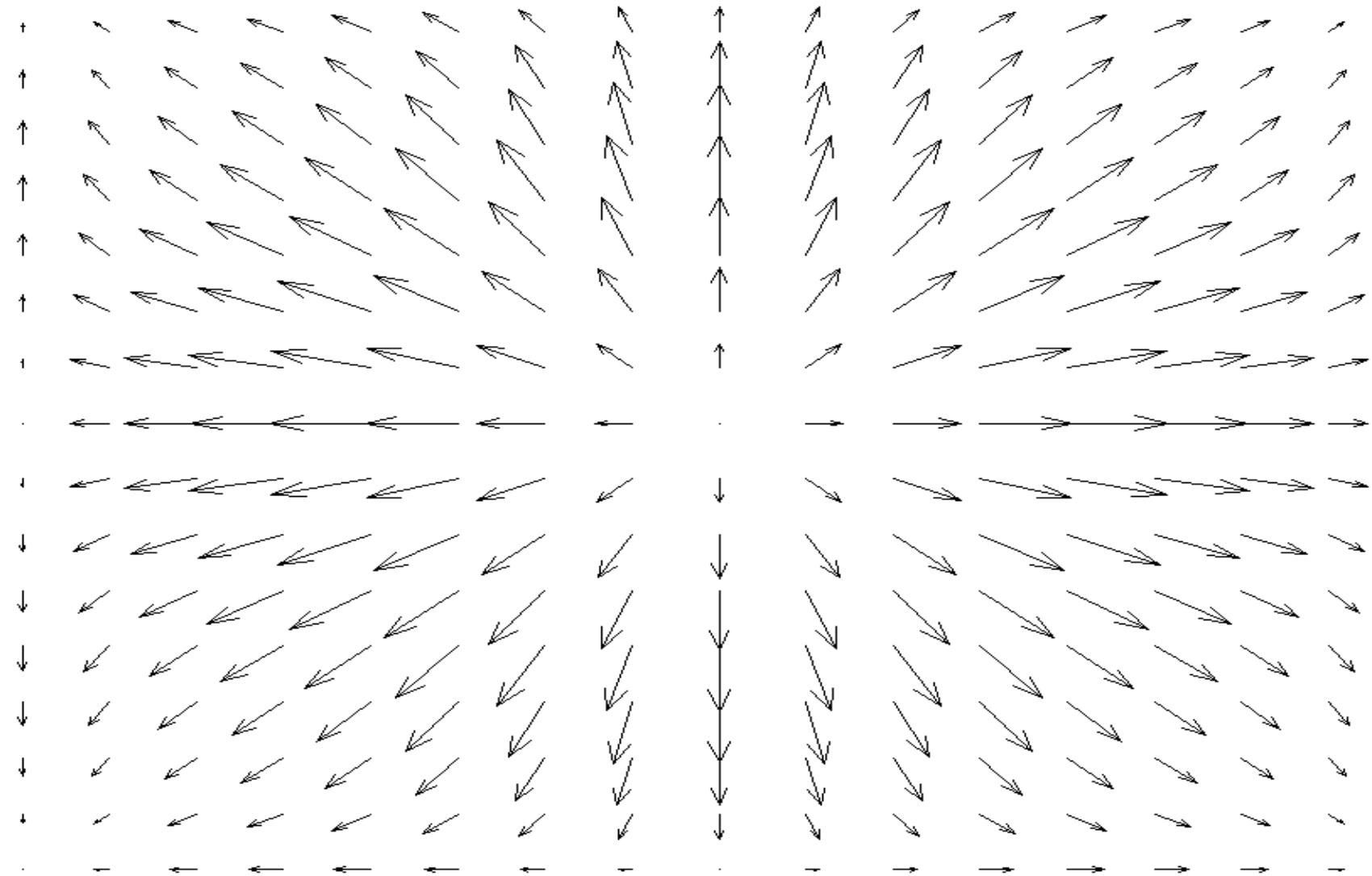
Alzheimer's



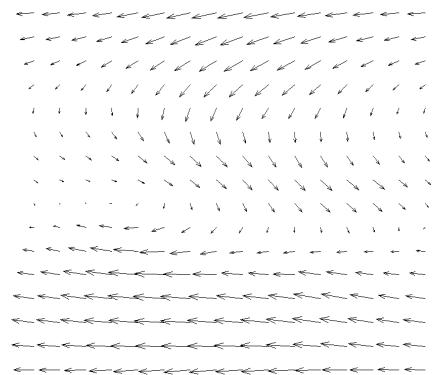
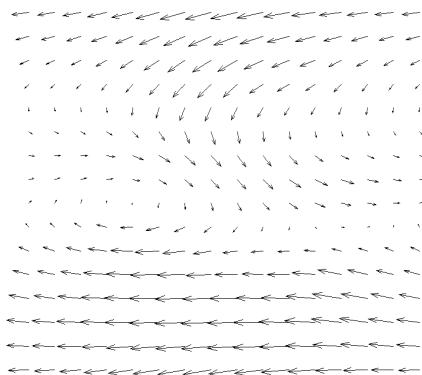
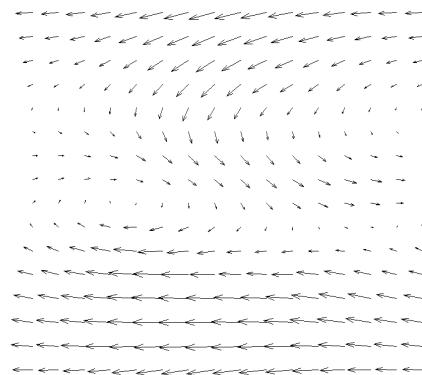
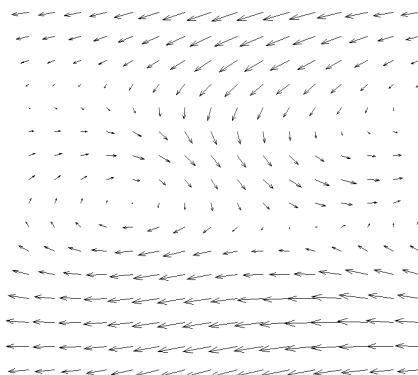
CIRCLE GEODESIC (scale)



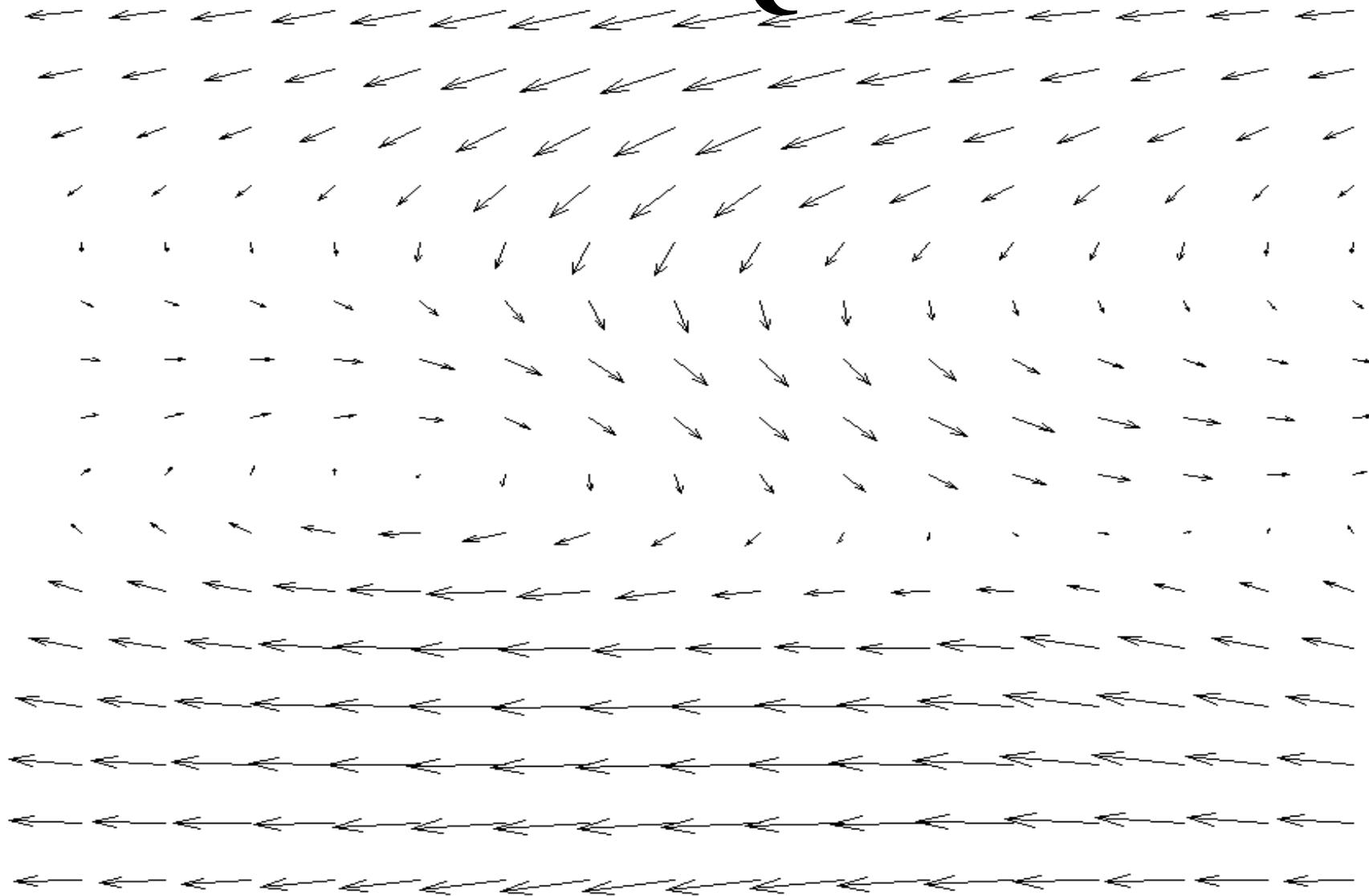
SCALE



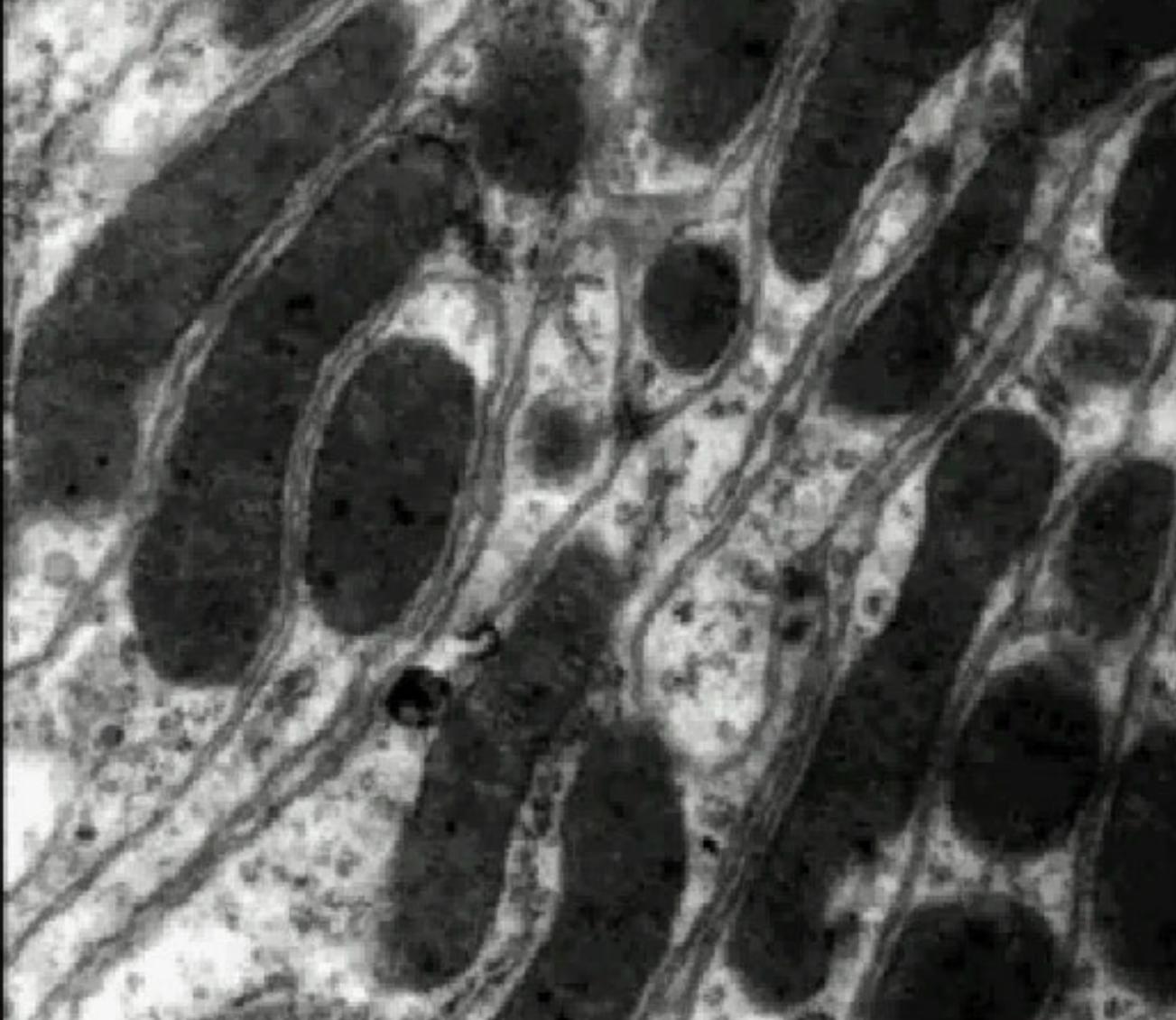
Macaque Geodesic



MACAQUE



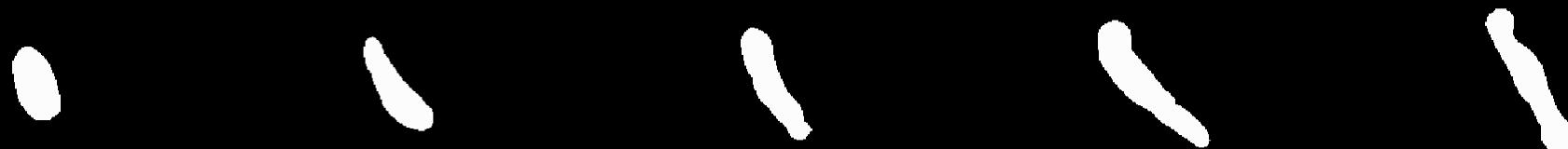
Cells: Scenes of Mitochondria



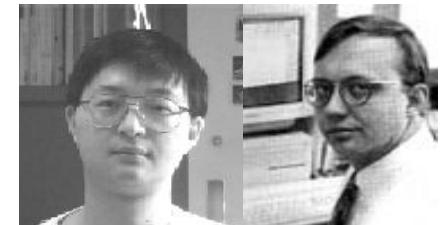
jump

The Metric Space of Mitochondria

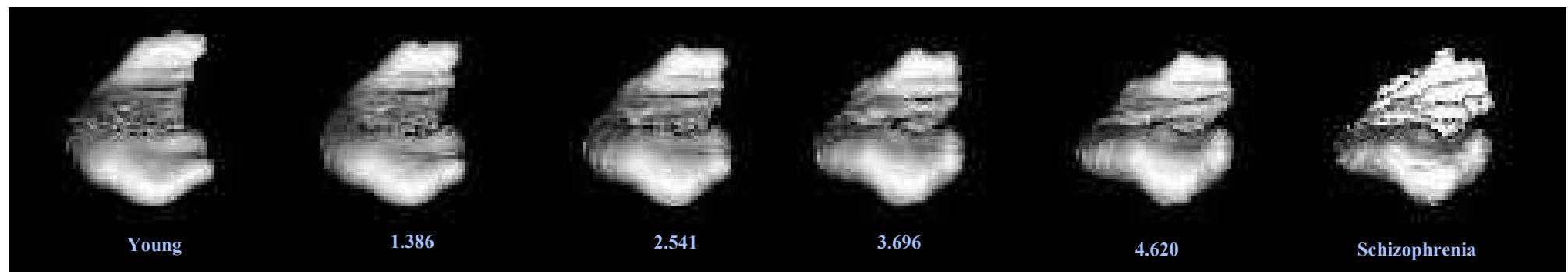
0 0.845 1.747 2.568 4.410



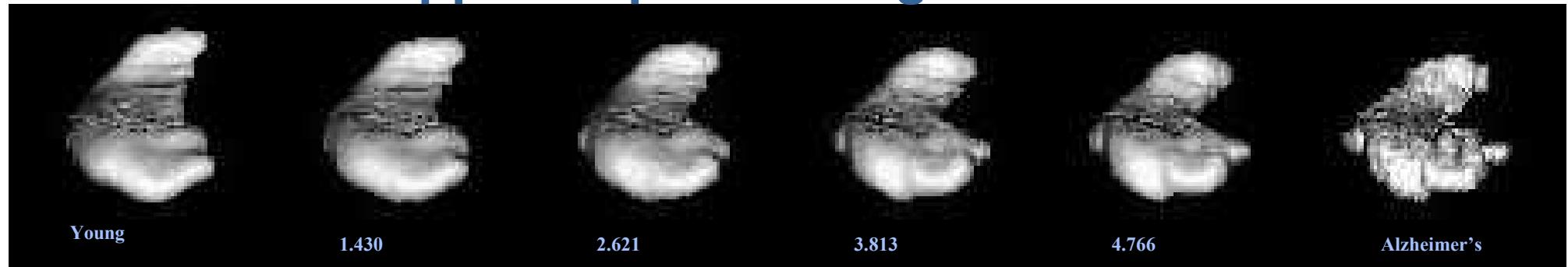
Metrics on 3D Hippocampus in Neuro-psychiatric Disorders.



3D Hippocampus: Young to Schizophrenia



3D Hippocampus: Young to Alzheimer's



Data from the lab. of Dr. Csernansky, Washington University, St Louis

Euler-Poincaré Equations of Computational Anatomy (1)



Baigent

- Mumford & Vishik (1998): E-L equations for exact matching
 - Mumford also showed that in 1D if $L=Id$, $\text{div } v = 0$, get Burgers' equation
 - Miller, Trouvé & Younes (2002): E-L equations for inexact matching
- $$\frac{\partial}{\partial t} (Lv(t))^t + (\nabla^t \cdot v(t))(Lv(t))^t + (v(t) \cdot \nabla^t)(Lv(t))^t + (Lv(t))^t \nabla^t v(t) = 0$$
- $$(Lv(1))^t + (I_0 \circ \phi^{-1}(1) - I_1) \nabla^t I_0 \circ \phi^{-1}(1) \nabla^t \phi^{-1}(1) = 0$$
- PDE can be considered as a generalized version of the Lagrangian Averaged Euler- α equations (cf. Camassa & Holm, 1993; Holm, Marsden & Ratiu, 1998; Marsden & Shkoller, 2001)

Euler-Poincaré Equations of Computational Anatomy (2)

- the E-L equations of CA can be written in the Euler-Poincaré form subject to the end-point condition at $t=1$ (transversality condition):

$$\frac{d}{dt} \frac{\delta l}{\delta v} = \text{ad}_v^* \frac{\delta l}{\delta v} \quad \text{where} \quad \frac{\delta l}{\delta v} = Lv$$

- parallels with the E-P analysis of the Camassa-Holm shallow water equation are interesting (e.g. Holm, Marsden & Ratiu, 1998)